fMRI pattern classification using neuroanatomically constrained boosting

Manel Martínez-Ramón,[a,c,*] Vladimir Koltchinskii,[b] Gregory L. Heileman,[a] and Stefan Posse[c]

[a]*Department of Electrical and Computer Engineering, University of New Mexico, NM 87131, USA*
[b]*Department of Mathematics and Statistics, University of New Mexico, NM 87131, USA*
[c]*The MIND Imaging Center, Department of Psychiatry, University of New Mexico, NM 87131, USA*

Pattern classification in functional MRI (fMRI) is a novel methodology to automatically identify differences in distributed neural substrates resulting from cognitive tasks. Reliable pattern classification is challenging due to the high dimensionality of fMRI data, the small number of available data sets, interindividual differences, and dependence on the acquisition methodology. Thus, most previous fMRI classification methods were applied in individual subjects. In this study, we developed a novel approach to improve multiclass classification across groups of subjects, field strengths, and fMRI methods. Spatially normalized activation maps were segmented into functional areas using a neuroanatomical atlas and each map was classified separately using local classifiers. A single multiclass output was applied using a weighted aggregation of the classifier's outputs. An Adaboost technique was applied, modified to find the optimal aggregation of a set of spatially distributed classifiers. This Adaboost combined the region-specific classifiers to achieve improved classification accuracy with respect to conventional techniques. Multiclass classification accuracy was assessed in an fMRI group study with interleaved motor, visual, auditory, and cognitive task design. Data were acquired across 18 subjects at different field strengths (1.5 T, 4 T), with different pulse sequence parameters (voxel size and readout bandwidth). Misclassification rates of the boosted classifier were between 3.5% and 10%, whereas for the single classifier, these were between 15% and 23%, suggesting that the boosted classifier provides a better generalization ability together with better robustness. The high computational speed of boosting classification makes it attractive for real-time fMRI to facilitate online interpretation of dynamically changing activation patterns.

Keywords: Functional magnetic resonance imaging; Pattern classification; Support vector machines; Adaboost

* Corresponding author. The UNM MIND Imaging Center, MSC11-6040, 1101 Yale Boulevard NE, Albuquerque, NM 87131-0001, USA.
E-mail address: manel@ieee.org (M. Martínez-Ramón).

Introduction

Brain activation changes in response to even simple sensory input and motor tasks encompass a widely distributed network of functional brain areas. Information embedded in the spatial shape and extent of these activation patterns, and differences in voxel-to-voxel time course, are not easily quantified with conventional analysis tools, such as statistical parametric mapping (SPM) (Kiebel and Friston, 2004a,b). Pattern classification in functional MRI (fMRI) is a novel approach, which promises to characterize subtle differences in activation patterns between different tasks. However, automatic and reliable classification of patterns is challenging due to the high dimensionality of fMRI data, the small number of available data sets, interindividual differences in activation patterns, and dependence on the image acquisition methodology. Recent work by Cox and Savoy (2003) demonstrated that linear discriminant analysis and support vector machines (SVM) allow 10-way discrimination of visual activation patterns evoked by the visual presentation of various categories of objects on a trial-by-trial basis within individual subjects. LaConte et al. (2003, 2005) used a linear SVM for online pattern recognition of left and right motor activation in single subjects. Wang et al. (2004) applied an SVM classifier to detect brain cognitive states across multiple subjects. Kamitani and Tong (2005) and Haynes and Rees (2005) used classification techniques applied to primary visual cortex to predict among different visual stimuli. In these papers, the dimensionality problem was not addressed, although better results in generalization ability can be achieved by reducing the dimensionality of the data. Jeswani and Posse (2004a,b) presented a method for reducing the resolution of the images to improve the generalization ability in linear maximum margin machines such as SVM.

All of the above work has been carried out using SVM (Vapnik, 1998; Burges, 1998). These algorithms have shown excellent performance in many applications (see http://www.kernel-machines.org for a collection of related papers). SVM, as well as boosting (Schapire, 1999), that was our major tool in this research, belong to so-called large margin classification methods.

In these methods, the parameters of classifiers are being tuned by minimizing the penalized empirical risk with respect to a convex loss function (also called in the literature margin cost function). This leads to convex optimization problems for which rather efficient algorithms have been developed. In particular, in the case of classical SVM, a simple piecewise linear loss (the hinge loss) is being used, the penalty being a quadratic functional. As a result, the penalized empirical risk minimization reduces to a quadratic programming problem for which there exists unique solution that can be found by very fast computational algorithms. The quadratic penalty is often defined as squared norm in a reproducing kernel Hilbert space (RKHS) associated with a symmetric nonnegatively definite kernel that defines the inner product. The choice of kernel might be crucial for the success of the method in complex classification problems. The most popular kernels are Gaussian radial basis and polynomial.

Boosting algorithms are a set of techniques for optimal aggregation of classifiers. These techniques use different base classifiers applied to the same data and then they find an optimal aggregation of the classifiers. The aggregated output has improved classification performance with respect to the best of the base classifiers. A boosting algorithm sequentially trains a set of classification machines that will be combined to produce an output. In each iteration, an aggregation parameter is computed based on the misclassification rate of the machine over a validation data set, so machines with lower misclassification rate will have a higher aggregation parameter. Prior to the training of the following machine, a distribution of weights over the training data is updated. If given data are misclassified, its weight will be increased, and if the data are well classified, its weight is decreased. This distribution is used during the training of the subsequent machine in such a way that the training will focus on those samples that have been difficult to classify by the previous one. It has been shown that provided enough data for training, weak classifiers (performing slightly better than random guessing) can be aggregated to form an arbitrarily good classifier satisfying any desired (classification or generalization) error tolerance. Schapire introduced the first polynomial time boosting algorithm in 1990, and in 1995, Freund and Schapire (1996) introduced the Adaboost algorithm for binary classification (see also Schapire, 1999), and, later, versions for multiclass classification (Allwein et al., 2000). Mason et al. (1999) introduced a general framework for Boosting design based on convex risk minimization with arbitrary convex loss function. A repository of published works on this topic can be found at http://www.boosting.org.

In the recent years, there has been significant progress in the understanding of generalization performance of large margin classification methods. Rigorous bounds on their generalization error have been proven and strategies for model selection based on the data (in particular, for selecting kernels and regularization parameters in SVM type methods) have been developed (see, e.g., Boucheron et al. (2005): http://www.econ.upf.es/lugosi/pubs.html).

Boosting techniques are particularly interesting for fMRI data which consist of high-dimensional data matrices and a small number of data sets. In order to reduce the dimensionality of the data without reducing image resolution, a new approach is presented that splits the activation maps into areas, applying a local (or base) classifier to each one. The dimension of each area will be smaller than the dimension of the whole activation map. Local classifiers are then trained on the data from each of the areas. An optimal aggregation of these local classifiers through boosting is able to select those areas in which useful information is present, discarding the others, and highly improving the classification performance with respect to the performance of the local classifiers. Also, the output of boosting in the form of boosting maps that highlights relevant activated areas can be directly compared with the output from conventional fMRI analysis that is reported as maxima of activation clusters. The strategy to split activation maps must be chosen carefully, since the performance of the classifier depends on it. In this work, prior knowledge of neuroanatomical parcellation of the brain into different functional areas (FA) for segmentation of activation maps has been used.

As local classifiers, the direct multiclass SVM algorithm has been applied (Weston and Watkins, 1999; Hsu and Lin, 2001). Two multiclass boosting approaches have been tested. In the first, the vector of outputs of each local classifier is linearly combined using a set of aggregation parameters. In the second, the outputs of each classifier are aggregated using an independent set of parameters, leading to as many aggregations as classes.

The data for training and classification consists of t-maps of the fMRI data acquired during the fMRI experiments. The use of t-maps implies the assumption that there is more information available about the temporal behavior of the task than there is spatial information about the activation patterns. As pointed by LaConte et al. (2005), this is, in general, not true. The temporal information is retrieved based on the use of a reference vector that indicates the intervals of time of the activity or stimulus, and on the use of a model of the hemodynamic response of the brain. The accuracy of the t-map depends on the accuracy of the reference vector and the hemodynamic response model. Thus, some quantity of information will be lost during the construction of the t-map. This is a common issue associated with any data reduction technique. Nevertheless, the techniques described here are not restricted to the use of t-maps and extensions to time series analysis of data are possible and are in fact under study.

Four-class classification accuracy was assessed in an fMRI group study with interleaved motor, visual, auditory, and cognitive task design across 18 subjects, and compared to conventional classification of the entire pattern, as well as classification of each pixel individually. The task design was such that activation patterns partly were shared between tasks and task duration was short, making the classification challenging. Classification accuracy with respect to different image acquisition methods was examined comparing data acquired with the same paradigm at 1.5 T and 4 T, using different image resolution and comparing multiecho EPI vs. conventional EPI. The results we present here are superior to those obtained using classical approaches to pattern classification.

Preliminary accounts of this work have been presented in Koltchinskii et al. (2005) for binary fMRI pattern classification and for multiclass classification in Martínez-Ramón et al. (2005a,b).

Materials and methods

Classification

The primary problem addressed in this paper is the high dimensionality of the data. Activation maps (i.e., t score or correlation coefficient maps) have about 90,000 active voxels, but the total number of available data for training purposes is about 100. In any case, the number of available activation maps will

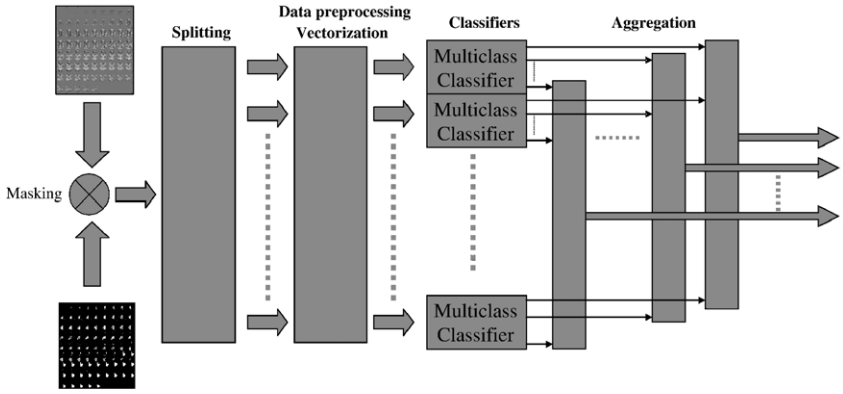

Fig. 1. Structure of the classifier.

always be much smaller than the number of voxels in one of them. This may preclude good generalization properties of the classifiers.

Thus, the first task in such a classification application is to reduce the dimensionality of the data. The first solution applied by several authors is the application of an analysis of variance (ANOVA) (Cox and Savoy, 2003) to discard those voxels which are considered to contain only noise. In addition, some authors suggest the reduction of the resolution as a way to reduce dimensionality. Nevertheless, reducing the resolution implies a low-pass filtering of the signal, so a possible loss of information can preclude good classification performance. Other techniques look for those dimensions which contain information. Principal Component Analysis (PCA) is widely used as a preprocessing for dimensionality reduction for single subject and group space-time source separation (e.g., Calhoun et al., 2001). But here are two additional problems. PCA needs to deal with square matrices which contain as many columns and rows as dimensions, the resulting computational burden makes the algorithm unsuitable for real-time implementations. On the other hand, this technique is only accurate if the total amount of data is large, but only small data sets are available in *t*-map classification tasks.

Our approach is based on the fact that the information in the brain is sparse, only a few areas of it will contain relevant information for the classification tasks. A reasonable segmentation of the activation map in anatomically constrained FA can be performed. Then, each area of the map will contain a smaller number of voxels. A local classifier applied to one FA which contains information relevant to the classification might show an improved generalization performance with respect to a classifier applied to the whole map. If the functional area does not contain information, the output of the corresponding classifier will show a nearly random performance. These behaviors can be automatically detected and then the good candidates selected and combined in an aggregation of classifiers by means of boosting techniques.

Structure of the classifier

The classifier consists of the following modules (Fig. 1).

Data reduction, normalization, and segmentation

Activation maps (e.g., *t*-maps or correlation maps) are normalized to unity. Then, the mean of the squared amplitude of each voxel is computed using all the training maps. Those voxels whose mean-squared value is less than the 30% of the maximum value among all voxels are removed. This procedure does not degrade the classification performance, and it reduces the total number of voxels used, thereby reducing the computation time.

In the following step, the activation maps are segmented into FAs based on the atlas of Talairach Tournoux areas (Talairach and Tournoux, 1988) using AFNI fMRI analysis software, available at http://www.afni.nimh.nih.gov/afni. A total of 14 masks were used to extract the FAs, grouped into left and right brainstem, cerebellum, parietal, temporal, occipital, subcortical, and frontal

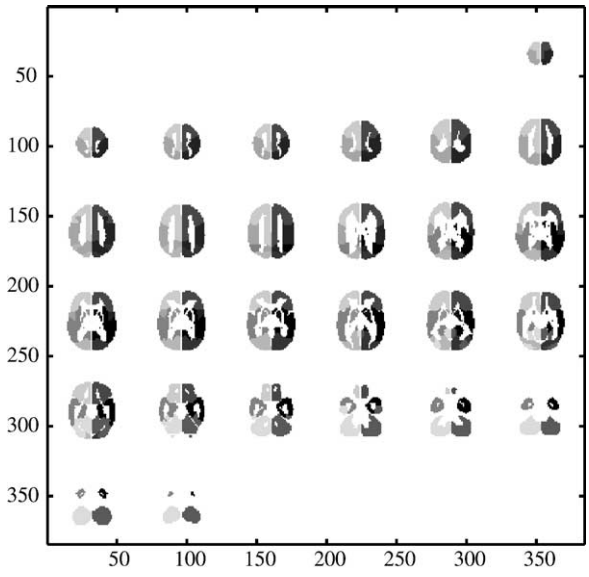

Fig. 2. Fourteen masks used to extract the functional areas.

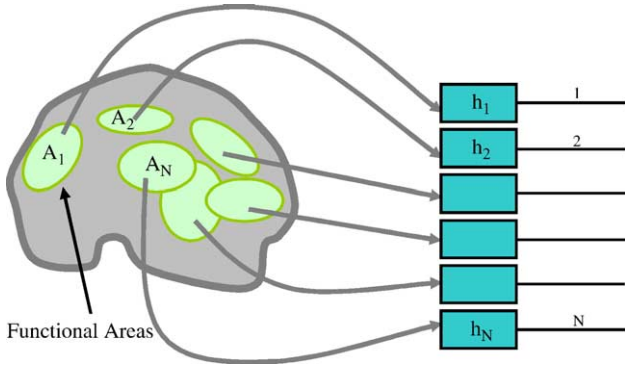

Fig. 3. Application of a local classifier for each of the 14 brain mask areas.

(Fig. 2). Additional masks including brainstem, cerebellum, parietal, temporal, occipital, subcortical areas, and frontal cortex were created using BRAINS2 image analysis software Magnotta et al. (2002) which can be obtained by request at http://www.psychiatry.uiowa.edu.

Classifier

A set of multiclass local classifiers $\boldsymbol{h}_j$, $1 \leq j \leq N$ are constructed and trained, one for each of the N FAs (Fig. 3). All classifiers provide a vector of L outputs, L being the number of classes. If the pattern has been classified as belonging to class l, output l of the classifier will be 1, and the remaining outputs will be -1.

SVMs are applied as local classifiers (Vapnik, 1998). The chosen schema for multiclass classification is the direct multiclass SVM classifier as described by Weston and Watkins (1999), which is experimentally more efficient than the classical one-against-one, one-against-all, error-correcting-output codes (Dietterich and Bakiri, 1995) or directed acyclic graphs (Platt et al., 2000) strategies.

Boosting block

The outputs of the local classifiers are linearly aggregated as

$$\boldsymbol{h} = \sum_{j=1}^{N} \lambda_j \boldsymbol{h}_j. \tag{1}$$

The aggregation is optimized by means of a boosting algorithm. The classifiers applied to FAs which have no information will have zero or low values in their aggregation parameter λ_j, as they will show poor classification performance. The classifiers which yield good performance will be boosted by increasing their aggregation parameter. As a result, a parameter λ_j will be assigned to each FA, which measures its importance for the given classification task. Then, a boosting map of the brain can be constructed highlighting those FAs containing relevant information for the classification.

Optimal aggregation of a set of classifiers

Boosting algorithms can be viewed as algorithms that perform empirical risk minimization with convex loss over a linear span of a given base class of functions (classifiers) using an iterative gradient functional descent method. This is typically implemented by maintaining a distribution of weights over the training set. At each iteration, the algorithm attempts to minimize the weighted training error over the base class and then it updates the weights of the training examples in such a way that those samples which are misclassified at the current iteration will have a higher weight at the next iteration. In this way, the learning machine will focus on the examples which are hard to classify at the current iteration. The algorithm also assigns nonnegative coefficients to the base classifiers obtained at each iteration and, at the end, outputs a convex combination of these classifiers (Fig. 4). It can be shown that this strategy indeed implements a version of gradient functional descent (see Appendices 1 and 2 for a more detailed description of the algorithm and Schapire, 1999 and Mason et al., 1999 for a detailed discussion of numerical and other aspects of boosting).

In our application, the base class for boosting consists of a finite number of local classifiers trained in advance in each of the FAs, and our method can be viewed as a "distributed" version of boosting, very natural in the classification of images. At each iteration, boosting picks one of the classifiers that minimizes the current weighted training error. The output of the algorithm is, in this case, a convex combination of local classifiers, which can be viewed as a method of their optimal aggregation to produce a classifier with the smallest risk. The coefficients of the convex combination show relative importance of local classifiers and corresponding FAs for a particular classification problem and they are used to construct a boosting map.

Assume that a set of pairs $\{\boldsymbol{x}_i, \boldsymbol{y}_i\}$ are available for classification tasks, where $y_i^{(l)}$ is 1 if the pattern belongs to class l, and -1 otherwise. In our fMRI pattern classification task, each

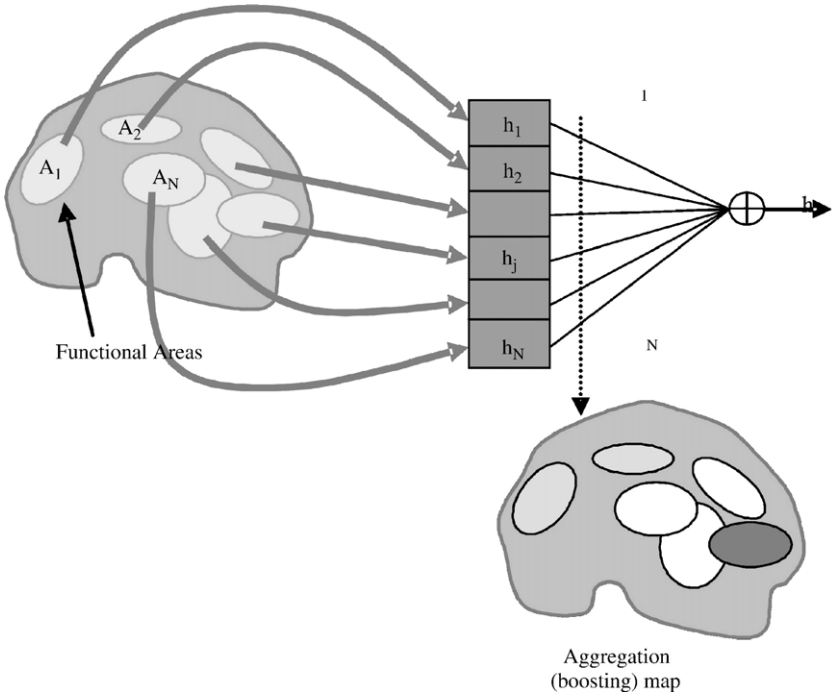

Fig. 4. Combination of classifier outputs to generate boosting maps.

pattern $\boldsymbol{x}_i$ is a vector constructed by sorting all the voxels of a t-map into a row. Each pattern $\boldsymbol{x}_i$ has a label $\boldsymbol{y}_i$. If the classification task involves more than two classes of t-maps, the labels associated with the patterns are vectors. In the classification tasks, there are four classes of patterns, corresponding to visual, motor, cognitive, and auditory. Arbitrarily, one can assign the label $\boldsymbol{y}_i = \{1, -1, -1, -1\}$ to the visual patterns, $\boldsymbol{y}_i = \{-1, 1, -1, -1\}$ to the motor ones, and so on.

The local classifiers will produce a vector output $\boldsymbol{h}_j = [h_j^{(l)} \cdots h_j^{(L)}]$ equal to one of the L possible labels. Here, the procedure of Allwein et al. (2000) is followed to construct a multiclass boosting algorithm. Two different strategies have been adopted and compared. Detailed descriptions of both algorithms are provided in Appendices 1 and 2). The first one is a natural extension of Adaboost to multiclass classification. This strategy can be found in Allwein et al. (2000) for standard Adaboost. Here the strategy is adapted to distributed Adaboost. The output of the algorithm is a convex combination of local classifiers as

$$\boldsymbol{h} = \sum_{j=1}^{N} \lambda_j \boldsymbol{h}_j \tag{2}$$

where λ_j is a scalar which weights each of the L-dimensional local classifiers. The second strategy is a modification to produce a set of L-dimensional parameters $\lambda_j = [\lambda_j^{(l)} \cdots \lambda_j^{(L)}]$ which weight the local classifiers as

$$\boldsymbol{h}^{(l)} = \sum_{j=1}^{N} \lambda_j^{(l)} \boldsymbol{h}_j^{(l)} \tag{3}$$

Here, a set of parameters $\lambda_{j,l}$, $1 \leq l \leq L$ have been updated to separately aggregate each of the L binary outputs of the local classifiers. As a result, L separate boosting maps have been obtained, one corresponding to each class.

This approach may provide better results in those situations in which some of the binary classifiers trained in the same FA present different performance depending on the class. Also, having as many boosting maps as classes, a better detection of the areas with relevant information for the classification of a given class may be achieved. Since we have to deal with a small number of training examples, the following randomization technique was useful. We split the training data at random into two subsets as follows: for each class, half of the data is randomly picked and put into one of the subsets, and the remainder is put into the other subset. Thus, both subsets have approximately the same fraction of patterns of each class as the original set. The first subset is used for training of local classifiers and the second is used for boosting the aggregation of these classifiers. This procedure is repeated independently a numerous times. The aggregation coefficients are then averaged. At the end, local classifiers are trained again based on the whole training data and an aggregate classifier is created by applying to them the average boosting coefficients.

As an alternative, a bootstrap algorithm can be applied to this task (see Efron and Tibshirani, 1998, or Rojo et al., 2002, were the bootstrap for SVM is introduced together with an application to classification in cardiology), but experiments show good performance using this simple algorithm.

Functional magnetic resonance imaging experiment

Subjects and paradigm

Ten healthy subjects were studied on a 1.5 T Siemens Sonata scanner and another 10 in a 4.0 T Brucker MedSpec Scanner. Informed consent based on institutionally reviewed guidelines was obtained prior to participation in the study. Stimuli were presented via MR-compatible LCD goggles and headphones (Resonance Technology Inc., Northridge, CA). The paradigm consists of four interleaved tasks: visual (8 Hz checkerboard stimulation), motor (2 Hz right index finger tapping), auditory (syllable discrimination), and cognitive (mental calculation). These tasks are arranged in a randomized block design (8 s per block), with a crosshair serving as baseline for a total of 132 s per scan (Fig. 5). The total duration for each condition was thus approximately 27 s. Visual stimulation consisted of 8 Hz reversing black and white checkerboards. Finger tapping in the motor task was paced with an auditory tone (1 kHz). Subjects were asked to tap with maximum extension of the finger onto a button-response pad (Cedrus corp., San Pedro, CA). During the auditory task, subjects listened to recorded syllables (i.e., "Ah, Ba, Ha, Ka, Ra") and pressed a button when they heard the target syllable "Ta" (25% of syllables). The cognitive task consisted of mental calculations. Subjects were asked to sum three aurally presented numbers and divide the sum by three, responding with a button press when the sum was divisible by three without remainder (50% of trials). Subjects were instructed to attend to each task with a constant effort across scans and field strengths.

Data acquisition

fMRI data were acquired using single-shot echo-planar imaging with TR: 2 s, TE: 50 ms, ip angle: 90°, matrix size: 64 × 64 or 32 × 32 pixels, FOV: 192 mm. Data with the 32 × 32 matrix were acquired with different bandwidth, either with 1200 Hz/pixel (LBW) or with 2400 Hz/pixel (HBW), which changes the degree of geometrical distortion and the signal-to-noise ratio. Slices were 6 mm thick, with 25% gap, 66 volumes were collected for a total measurement time of 132 s per run. The available data set consists of 184 t-maps taken from 18 different subjects. Details of the data set are provided in Table 1.

Data analysis

The computer hardware used in all experiments consisted of an Intel Radeon at 3.2 GHz workstation with 512 Cache Memory running in Linux, and Matlab 6.5 (The MathWorks, Inc) engine with ANSI C code for the intensive computations (learning of the

Fig. 5. Representative visual stimuli used in interleaved paradigm.

Table 1
Number of t-maps acquired with different field strengths, spatial resolutions, and read-out bandwidths

Field	# t-maps	Resolution	# t-maps	BW	# t-maps
1.5 T	101	32 × 32	76	LB	55
				HB	21
		64 × 64	25	LB	25
4.0 T	81	32 × 32	52	LB	47
				HB	5
		64 × 64	29	LB	29

multiclass classifier parameters) combined. Statistical parametric mapping using SPM2 (Kiebel and Friston, 2004a,b) was performed to generate t-maps that represent brain activation changes. Preprocessing steps included motion correction, slice-time correction, spatial normalization, and spatial smoothing. Statistical analysis using a design matrix with four conditions (motor, visual, auditory, cognitive) was performed with corrected amplitude threshold ($P < 0.05$) and 132 s high-pass filter. Examples of the obtained t-maps in Fig. 6 show in part overlapping activation patterns due to auditory stimulus presentation in motor, cognitive, and auditory tasks, and button responses in motor, cognitive, and auditory tasks. In order to measure the performance of the classifier, the following tests were conducted:

(1) Dynamics of the boosting parameters. In order to test the time behavior of the Adaboost, we ran both distributed Adaboost algorithms I and II for all the 1:5 T data and looked at the behavior of the aggregation parameters and error rates.
(2) Misclassification results for cross modality training. Different tests were run to measure the cross modality performance of the classifier:
   - Training with all 1.5 T (4.0 T) t-maps, test with the 4.0 T (1.5 T) t-maps. The number of 1.5 T t-maps is 101, where there are 81 4.0 T t-maps.

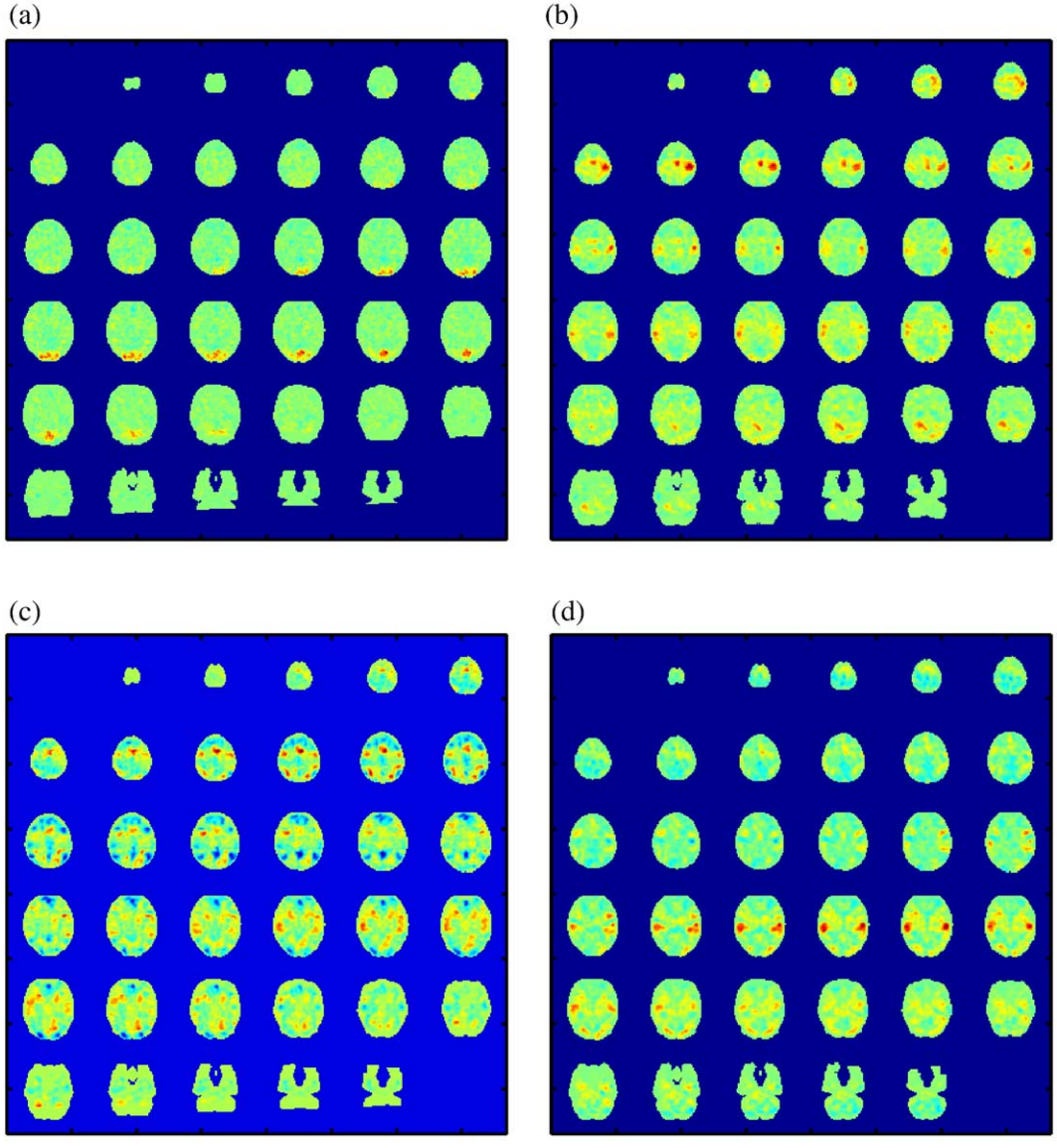

Fig. 6. Example of activation t-maps corresponding to visual (a), motor (b), cognitive (c), and auditory (d) activations in a single subject.

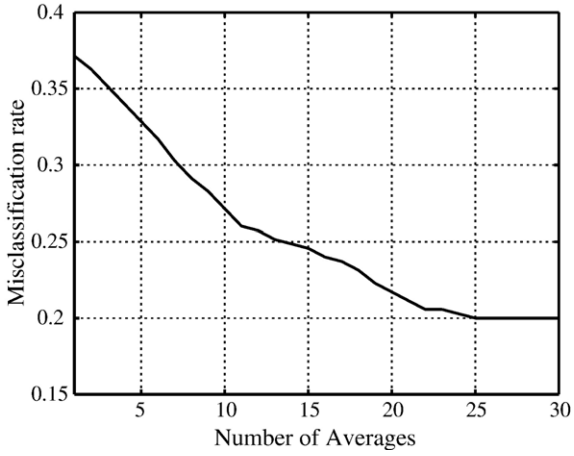

Fig. 7. Evolution of the test misclassification rate as a function of the number of averaged Adaboost parameters using the distributed Adaboost algorithm II and RBF-SVM local classifiers. Thirty different realizations of the experiment have been averaged to obtain this graph.

- Training with high (low) resolution and test with low (high) resolution t-maps. There are 128 low resolution t-maps and 54 high resolution t-maps.
- Training with low (high) bandwidth and test with low (high) resolution t-maps. There are 156 low bandwidth t-maps and 26 high bandwidth t-maps.

(3) Test with all data. We performed a leave-one-out test using all the available activation maps. This procedure consisted of training the algorithm with 218 maps, leaving one out for testing, which results in 219 different trainings and tests.

(4) Relative contribution of different classes to mixed activation patterns. The interleaved task design in the present study leads to a partial overlap of activation patterns from different classes, and the classifier assigns multiple labels in accordance with the relative contribution of different activation patterns. It can be said that the activation maps used in the experiments are often multilabeled, that is, they belong to more than one class.

Provided that the activations belonging to different tasks have different locations in the brain, the introduced classification method has some ability to detect multilabel patterns, because it consists of the aggregation of the outputs of local classifiers that work based on different brain areas.

In order to detect different classes of activations, instead of just aggregating all of the local classifier outputs, one may group the classifiers whose outputs belong to each class and aggregate them separately using the obtained aggregation parameters of algorithm II. One can define the following quantities:

$$o_l = \frac{\sum_{i:h_{i,l}=1} \lambda_{i,l}}{\sum_{l=1}^{L} \sum_{i:h_{i,l}=1} \lambda_{i,l}}, 1 \le l \le L \tag{4}$$

that represent the fraction of detected activation for each class. Here, the aggregation parameters weight the predicted labels of each classifier. This is a weighting of the predicted labels: those classifiers which have shown poor classification performance during the training of the boosting aggregation will have low valued aggregation parameters, so their predicted label will be weighted with a low value, and the ones with good performance will be weighted with a higher value. This can be viewed as that the "best" classifiers are given a better credibility in the classification. The output will not only be a label, but also a vector of L scalars, each one with a weight indicating the fraction of each predicted activation.

Results

Dynamics of the boosting parameters

The local classifiers consisted of SVMs with Gaussian RBF kernel function. The Adaboost training was repeated up to 100 times and the resulting sets of aggregation parameters were averaged and normalized to 1, as described in Optimal aggregation of a set of classifiers. Specifically, in each training of Adaboost, 25 t-map from the training data set are randomly picked and used to train the local classifiers. The other 25 are used to train the parameters.

Fig. 7 shows the error rate as a function of the number of averaged Adaboost parameters obtained from algorithm II. To compute the error rate, the local classifiers are trained with the entire training data set and then the aggregation is tested with the averaged Adaboost parameters. The error rate as a function of time corresponding to algorithm I shows the same behavior. We can see in this figure that the performance increases with the number of averages (although, with the current amount of data, there is no statistical significance on the improvement). As the amount of data is small, the resulting aggregation parameters from each Adaboost training are inaccurate, but for this test set, averaging 25 aggregation parameters is enough to obtain stable results.

Fig. 8 shows the behavior of the aggregation parameters of algorithm II from the previous experiments. Each graph in the figure is the average over l of all parameters $\lambda_{j,l}$. The parameters are quite stable after 40 iterations, although the results of Fig. 7 suggest that 20 to 25 iterations may be enough to achieve good generalization performance for this experiment.

Fig. 9 shows the boosting maps obtained using algorithm I and the average of the four obtained boosting maps with

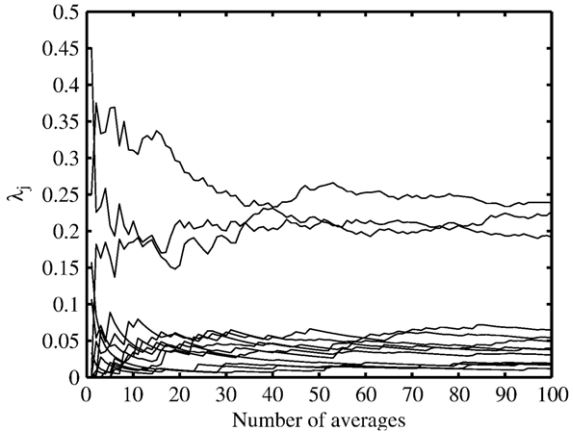

Fig. 8. Evolution of the aggregation parameters λ as a function of the number of randomized boosting iterations using the distributed Adaboost algorithm II and RBF-SVM local classifiers. Each graph is the average over l of all parameters $\lambda_{j,l}$.

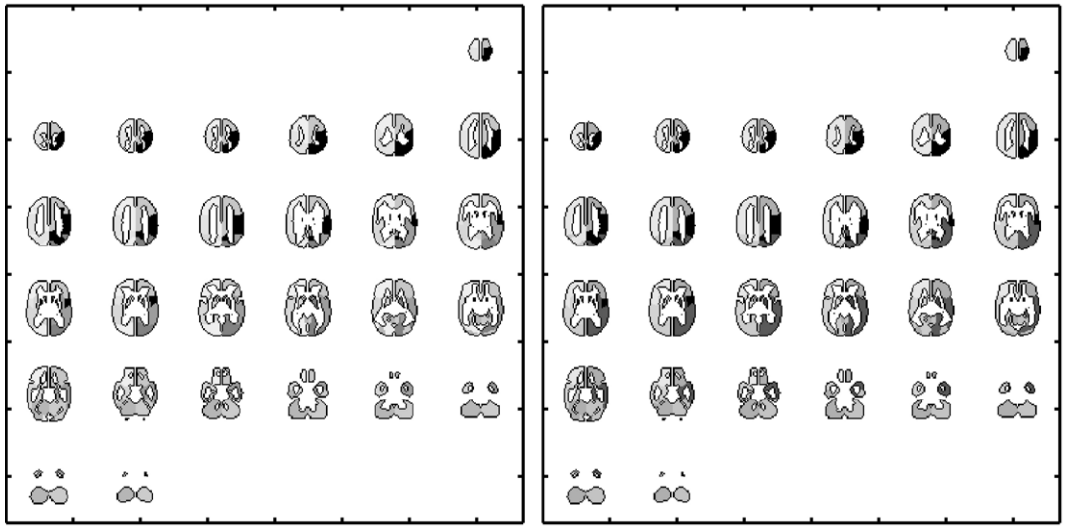

Fig. 9. Left: boosting map obtained with the distributed Adaboost algorithm I. Right: average of the four boosting maps II shown in Fig. 10 using the algorithm. Darker areas correspond to higher aggregation parameter values.

algorithm II. We can see that both maps are quite similar, as can be expected. The most important amplitudes are in right parietal, subcortical, and temporal lobes, followed by left occipital and right frontal lobe.

Fig. 10 shows the four boosting maps produced by algorithm II. The visual map does not show high values exclusively in the occipital areas because all areas contain information which can be used for classification. For example, the right subcortical and temporal areas can be used for the classification due to the fact that there is no activation in them during the visual activity but there is during all other activities. On the other hand, activity in the cerebellum is present during motor task, which extends partly into the occipital cortex due spatial normalization and smoothing. The activation map corresponding to motor activation shows a high value in the right parietal area, where the motor activation is present. For the case of the cognitive and auditory maps, the right temporal area is relevant for the classification, but not the left one; it typically contains more noise or interleaved activation in the images used for training, which precludes its use for classification. Interestingly, the occipital areas are used for classification of the cognitive activity Fig. 11 show the classification given by all local classifiers for all test activation maps, plus the decision given by the boosting aggregation of algorithm II. The four colors represent the four possible answers of the classifiers: dark blue for visual, clear cyan blue for motor, yellow for cognitive, and red for auditory. Each row of the graph corresponds to all of the classifier outputs for an activation map, sorted as left and right brainstem, cerebellum, frontal, occipital, parietal, subcortical, and temporal. Each column shows the output of one classifier for all of the activation maps. We sorted the activation maps according their label, first the visual, then motor, cognitive, and finally auditory. The last column (into the black frame) shows the aggregated output for all activation maps. It can be seen that the aggregation shows the best performance, with only three misclassified cognitive activations.

Misclassification results for cross modality training

The results of the tests are shown in Table 2. We can see that in all cases distributed Adaboost performs better than the single classifier. This might be related to the fact that the boosting classifier has a special structure. Specifically, it is a convex combination of local classifiers related to specified functional areas of the brain. In the case when there are only few functional areas that contain relevant information for a particular classification problem, the convex combination becomes sparse, which leads to a reduction in the complexity of the classifier and the improvement of its generalization performance. Global SVM classifiers cannot achieve this goal unless they are based on a specially designed kernel that takes the functional areas into account. Because of this, global SVMs tend to overfit as compared to distributed boosting.

Algorithm II shows a performance slightly better than algorithm I in four of the tests, worse in one and equal in one. The better performance of algorithm II may be explained by the fact that it applies four Adaboosts to the four outputs of the classifiers. Then, the algorithm can use a classifier that shows good performance in the classification of a given class even if it is random for the other classes, while algorithm I cannot. Nevertheless, the differences are very small given the small data sets that we are using. In fact, in the worst cases, algorithm I misclassified two more activation maps than algorithm II.

It is interesting to note that the single classifier shows a slightly different behavior. Results are better in the tests across field strengths than those in the tests across resolutions and bandwidths, even if the number of training samples was higher when training with low resolution and bandwidth. This suggests that the single classifier may be more sensitive to differences in spatial resolution and geometrical distortion, as a function of readout bandwidth.

Almost all misclassifications were in auditory activation classification, where the information is not as sparse as in visual or motor activations, and where other kinds of activations were present, mainly cognitive and motor. As an example, Table 3 shows the confusion matrix for the experiment of 4 T vs. 1.5 T data. Among the 20 auditory activations, only 15 were correctly classified, where 5 of them were classified as cognitive. All visual and cognitive activations were correctly classified and only 1 out of 23 motor activations were misclassified. The table also shows

the result of the high bandwidth vs. low bandwidth test, with similar behavior.

The same set of tests was run for linear and second degree polynomial kernels. In both cases, results were poorer than with Gaussian RBF kernels. Linear classifiers produce poor performance due to the fact that the optimum separation hyperplanes are not linear. For polynomial kernels, we observe high sensitivity of the classification results to small changes in the parameters. For example, slight differences in amplitude scaling or order of the set of polynomials produce very different results, which make polynomials not well-suited for this kind of classification task.

Tests with all data

Results give a misclassification rate less than 2% for the boosting scheme, and above 9% for the single classifier, which suggests that the boosting scheme has better generalization ability.

Relative contribution of different classes to mixed activation patterns

Visual activation maps were very distinct, but most of the other activation maps show mixtures of activation patterns. As an example, the outputs described in Eq. (4) were computed for the activation maps in Figs. 6 and 12. The activation *t*-maps in Fig. 6 do show very distinct activation patterns, and are thus clearly classified as belonging to a single class. The maps in Fig. 12 obtained from a different subject show mixed activation patterns that are classified as belonging to multiple classes. Table 4 contains the fraction of overlapping tasks estimated by the classifier. For example, in the motor activation pattern in Fig. 12, there is an estimation of cognitive activity of 32%, which can be explained mainly by the activation in the center of the frontal and temporal areas, which is much less intense in the corresponding motor activation in Fig. 6. Also, one can see activation in the parietal area and cerebellum in the auditory

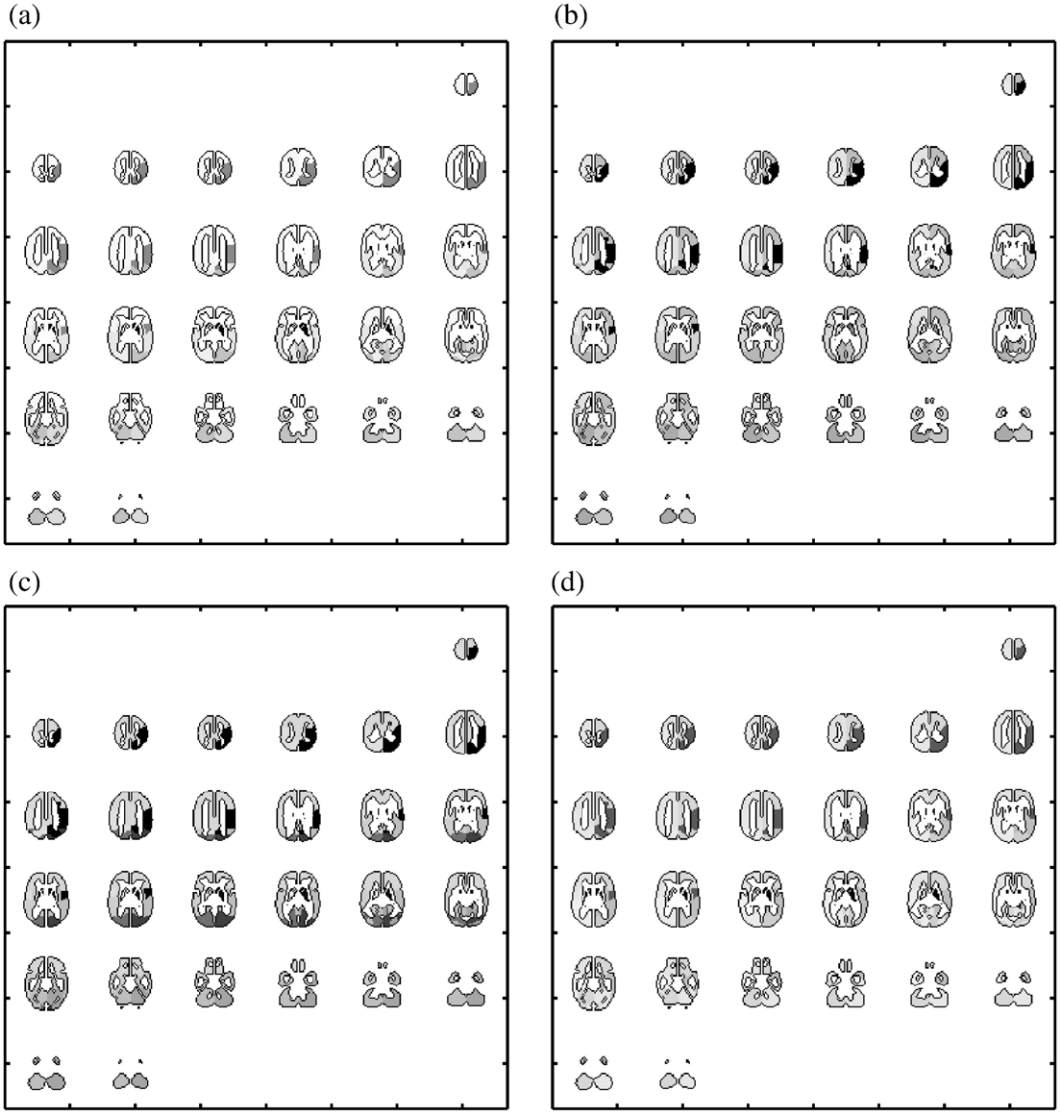

Fig. 10. Boosting maps for visual (a), motor (b), cognitive (c), and auditory (d) activation using the distributed Adaboost algorithm II and RBF-SVM local classifiers highlight areas that are important for classification with respect to other 3 conditions.

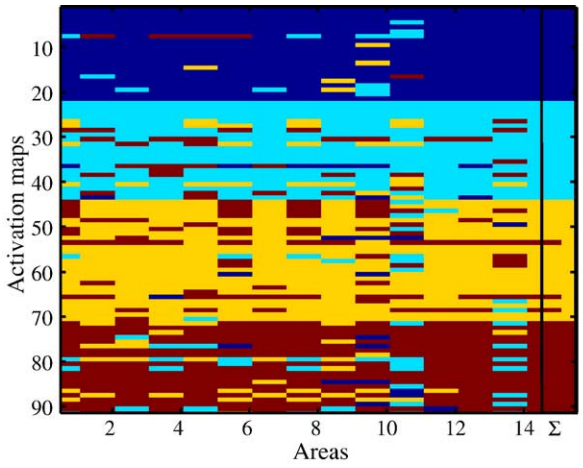

Fig. 11. Map of the output of all classifiers and the aggregation of them (∑) obtained using algorithm II for all test data. The four colors represent visual, motor, cognitive, and auditory classifications.

map in Fig. 12, which suggests motor activation, as detected by the classifier.

Discussion

We introduced a new classification method for application to multiclass classification of different activations in fMRI *t*-maps. The method is based on the segmentation of *t*-maps into different neuroanatomical areas, and the application of independent local classifiers to each one. The outputs of the classifiers are optimally aggregated using a spatially distributed version of the well-known Adaboost algorithm. The spatially distributed Adaboost chooses the areas with information relevant for the classification, discarding the rest, which makes the classifier more sparse and reduces its complexity. The reduction of the complexity makes the method robust across cross modality training. Also, we observed that the method is robust against variations in the choice of the parameters of the local classifiers. This is interesting for real applications, as the user does not need to interact with the algorithm to adjust parameters. In addition, the spatially distributed Adaboost produces a boosting map of the brain. This map is obtained from the amplitudes of the aggregation parameters and helps to highlight those areas of the brain with information relevant for the classification.

We tested the algorithm across different subjects, field strengths, resolutions, and bandwidths. We compared the results against those of standard classifiers, showing improved classification accuracy. The most similar results are obtained with tests of different field strengths. This is probably due to the fact that both data sets have a similar number of *t*-maps. Also, we can observe that the classification accuracy is high compared to others in the same table. Nevertheless, we cannot conclude that this is due to the differences between field strengths. The second and the third experiments have similar results, which are mainly due to the amount of data used for training and testing. When the structure is trained with low resolution or low bandwidth data, the number of *t*-maps for training is high (128 and 156, respectively). The test accuracies are the lowest, and they are similar. This suggests that training with different data does not make any difference, and that accuracy is mainly a matter of the number of training *t*-maps. For example, if we randomly choose 50 data points of low resolution to train, and then test the network at high resolution, the average misclassification rate is about 10%, similar to the fourth and sixth row of Table 4.

Table 2
Misclassification rates for single classifier and boosting classifier (algorithm I and algorithm II) using Gaussian Radial Basis Function kernels

Tr:	4 T	1.5 T	32 × 32	64 × 64	HB	LB
Test:	1.5 T	4 T	64 × 64	32 × 32	LB	HB
Alg. I	9.9%	8.6%	**1.9%**	9.4%	**3.9%**	11.5%
Alg. II	**7.9%**	**7.4%**	3.7%	**8.6%**	**3.9%**	**9.6%**
Single	14.8%	14.8	21%	23.3%	19%	23.1%

Boosting algorithm II shows slightly higher performance than Boosting algorithm I (best result shown in bold).

Table 3
Confusion matrices for: (left) training with 4 T data and testing with 1.5 T data, and (right) training with high read-out bandwidth and test with low read-out bandwidth data

4 T vs. 1.5 T					HBW vs. LBW				
	1	2	3	4		1	2	3	4
1	14	0	0	0	1	36	1	0	0
2	0	23	0	1	2	0	38	0	0
3	0	0	23	0	3	3	1	37	4
4	0	0	5	15	4	0	1	5	30

Algorithm II and Gaussian RBF kernel were used for local classifiers. Columns: actual class; rows: predicted class. 1: visual; 2: motor; 3: cognitive; 4: auditory.

The method has many potential uses in neuroanatomical studies. Rather than using the method to classify previously known activations, which are explicit in the paradigms used to compute the *t*-maps, we intend to predict hidden variables. These hidden variables may be related to the prediction of diseases, brain computing interfaces, and others. It is well suited for real-time applications, as it has a fast classification response and a fast training. We have recently interfaced the classifier to our real-time fMRI analysis software (TurboFIRE) (Gembris et al., 2000; Gao and Posse, 2004; Posse et al., 2001, 2003) (see also http://www.mic.health.unm.edu/turbofire) and are currently evaluating training and testing performance with cognitive paradigms. The limitations of the method are related to the sparsity of the information in the brain. If the information relevant for the classification is distributed in a large area of the brain, the performance of the classifier will degrade. In those cases, it may be better to apply a single classifier to the whole brain *t*-map. In addition, the method requires a good choice of neuroanatomical areas. A bad choice may result in a decreased performance. Nevertheless, there is no limitation to the number and extent of the applied neuroanatomical areas. They may even overlap, and one can leave the choice of the adequate areas to the distributed Adaboost procedure.

In this paper, we only consider the case in which the boosting technique is used to determine the relevance that particular areas should have in classification. It is possible that an advantage could be gained by taking into account the dependencies that exist between these areas, and to also use this information during

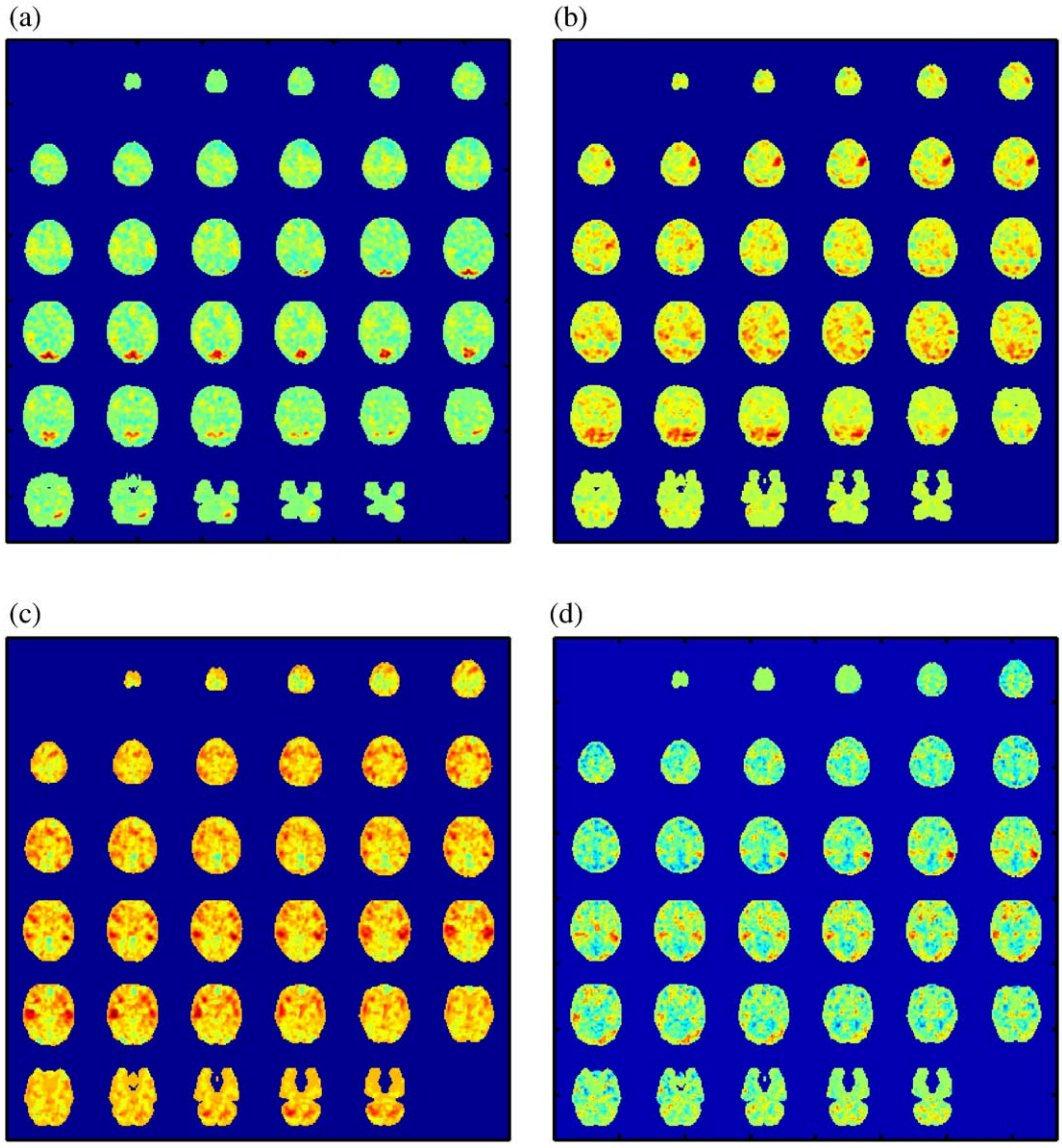

Fig. 12. Examples of activation *t*-maps corresponding to visual (a), motor (b), cognitive (c), and auditory (d) activations in a different subject showing mixtures of activation patterns.

training. The aggregation techniques that we use in this paper could be used to take advantage of the interactions between areas, as they can be applied to a more sophisticated set of classifiers that include classifiers trained on pairs or larger groups of functional areas. We are exploring several alternatives that can be used to design these groupings of areas. One of these alternatives involves using prior knowledge about the connectivity of the different brain areas. Another involves studying the dependencies between the outputs produced by classifiers applied to different functional areas, and then grouping according to this analysis.

Work in progress also includes other boosting approaches, such as logistic Adaboost, or applying multilabel local classifiers in order to detect different activations in the same activation map. We are also investigating combining activation patterns from different areas using boosting as mentioned in the Introduction. Other work in progress includes strategies to choose good anatomical masks, and including prior knowledge in boosting in order to speed up and improve the resulting boosting maps. We are currently expanding the classifier method to analyze the spatial-temporal dynamics in fMRI time series data.

Table 4
Relative contributions of task mixtures in the activation maps shown in Figs. 6 and 12

t-map	Visual	Motor	Cognitive	Auditory
Fig. 6a	1	0	0	0
Fig. 12a	0.91	0.06	0.03	0
Fig. 6b	0	1	0	0
Fig. 12b	0	0.67	0.32	0
Fig. 6c	0	0	1	0
Fig. 12c	0	0.14	0.58	0.28
Fig. 6d	0	0	0	1
Fig. 12d	0.01	0.24	0.03	0.72

Acknowledgments

We want to thank H. Jeremy Bockholt and Andrew Mayer (the MIND Institute, UNM, USA) for providing the brain masks and helping us to use them; Daniel Fitzgerald (Wayne State University School of Medicine), Kunxiu Gao, Jing Xu, and Ting Li (the MIND Institute) for expert technical assistance with scanning and data preprocessing; Jason Weston and Gökhan Bakir (Max Plank Institüt) for sharing their machine learning library Spider. This work was supported by NIH Grant NIBIB 1 RO1 EB002618-01, the MIND Institute-Mental Illness and Neuroscience Discovery DOE Grant No. DE-FG02-99ER62764, and NSF Grant DMS-0304861, Dept. of Mathematics and Statistics.

Appendix A. Distributed Adaboost (algorithm I)

The steps are:

- Initialize an error distribution matrix $D_0(i,l) = (1 / nL)$ for each data $\boldsymbol{x}_i$ and each class, and initialize a set of aggregation parameters $\lambda_{j,0} = 0$
- Repeat for $t = 1 \ldots T$
  - For each classifier, compute the classification error $\varepsilon_t(j)$

$$\varepsilon_t(j) = \sum_{l=1}^{L} \sum_{i=1}^{n} D_t(i,l) I\left\{ h_j^{(l)}(\boldsymbol{x}_i) \neq y_i^{(l)} \right\} \tag{A–1}$$

  where $I\{h_j^{(l)}(\boldsymbol{x}_i) \neq y_i^{(l)}\}$ is 1 if $h_j^{(l)}(\boldsymbol{x}_i) \neq y_i^{(l)}$ and 0 otherwise.
  - Choose the best classifier or the classifier which produces the lowest error. We denote the best classifier with the index $\hat{j}$.
  - Compute an update term α_t for the aggregation parameter corresponding to the best classifier

$$\alpha_t = \frac{1}{2} \ln\left(\frac{1 - \varepsilon_t(\hat{j})}{\varepsilon_t(\hat{j})} \right) \tag{A–2}$$

  and update the aggregation parameter

$$\lambda_{\hat{j},t+1} = \lambda_{\hat{j},t} + \alpha_t \tag{A–3}$$

  - Update the error distribution:

$$D_{t+1}(i,l) = \frac{D_t(i,l)}{2\sqrt{\varepsilon_t(\hat{j})\left(1 - \varepsilon_t(\hat{j})\right)}} \exp\left(- \alpha y_i^{(l)} h\hat{j}^{(l)}(\boldsymbol{x}_i) \right) \tag{A–4}$$

  where $2\sqrt{\varepsilon_t(\hat{j})\left(1 - \varepsilon_t(\hat{j})\right)}$ in the denominator is the normalization factor such that $\sum_i \sum_l D_{t+1}(i,l) = 1$.
- End.
- Normalize the aggregation set so that $\sum_j \lambda_j = 1$.

In order to stop the iterations, a normalized version of the parameters $\lambda_{j,t}$ may be computed in each iteration. If the variation of the normalized parameters is less than a threshold γ, the algorithm can be stopped. This is equivalent to stopping the algorithm when $\frac{\alpha_t}{\sum_j \lambda_{j,t}} < \gamma$.

Note that the update (A-2) tends to zero if the error tends to 1 = 2 (random output) and tends to infinity if the error tends to zero. So, the better the classifier performs, the higher its update is. On the other hand, Eq. (A-4) for the weight update makes the weights $D_t(i,l)$ corresponding to data $\boldsymbol{x}_i$ to grow if the classifier misclassifies it (as the exponent will be positive). If the classification is good, the weight will decrease. Then, in the next step, error measure (A-1) will give more importance to those data which are hard to classify.

It is well known that the above algorithm essentially minimizes the empirical risk with respect to exponential loss function by a gradient functional descent (see, for example, Mason et al., 1999).

The difference between Adaboost and the present procedure is that, all classifiers are already trained, and the distribution $D_t(i,l)$ is used to update the value of $\lambda_{j,t}$ rather than to train the next classifier (the best one is picked from a pretrained set of local classifiers).

Appendix B. Distributed Adaboost (algorithm II)

Here L binary distributed Adaboost algorithms have been applied to each one of the outputs of the base classifiers. Then, this can be viewed as a binary reduction of the first algorithm. As it is pointed before, the result is a set of L boosting maps corresponding to each one of the classes. The procedure is the following one.

For each j,l, do:

- Initialize an error distribution matrix $D_0(i,j,l) = (1/n)$ for each data $\boldsymbol{x}_i$ and each class, and initialize a set of aggregation parameters $\lambda_{j,0}^{(l)} = 0$.
- Repeat for $t = 1 \ldots T$
  - For each classifier, compute the classification error $\varepsilon_t(j,l)$

$$\varepsilon_t(j,l) = \sum_{i=1}^{n} D_t(i,j,l) I\left\{ h_j^{(l)}(\boldsymbol{x}_i) h_i^{(l)} \right\} \tag{A–5}$$

  where $I\{h_j^{(l)}(\boldsymbol{x}_i) \neq y_i^{(l)}\}$ is 1 if $h_j^{(l)}(\boldsymbol{x}_i) \neq y_i^{(l)}$ and 0 otherwise.
  - Choose the best classifier or the classifier which produce the lowest error.
  - Compute an update term $\alpha_t^{(l)}$ for the aggregation parameters $\lambda\hat{j}^{(l)}$ corresponding to the best classifier.

$$\alpha_t^{(l)} = \frac{1}{2} \ln\left(\frac{1 - \varepsilon_t(\hat{j},l)}{\varepsilon_t(\hat{j},l)} \right) \tag{A–6}$$

  and update the aggregation parameter

$$\lambda_{\hat{j},t+1}^{(l)} = \lambda_{\hat{j},t}^{(l)} + \alpha_t^{(l)} \tag{A–7}$$

  - Update the error distribution:

$$D_{t+1}(i,j,l) = \frac{D_t(i,j,l)}{2\sqrt{\varepsilon_t(\hat{j},l)\left(1 - \varepsilon_t(\hat{j},l)\right)}} \exp\left(- \alpha y_i^{(l)} h\hat{j}^{(l)}(\boldsymbol{x}_i) \right) \tag{A–8}$$

- End.
- Normalize the aggregation set so that $\sum_j \lambda_j^{(l)} = 1$.

References

Allwein, E.A., Schapire, R.E., Singer, Y., 2000. Reducing multiclass to binary: a unifying approach for margin classifiers. J. Mach. Learn. Res. 1, 113–141.

Boucheron, S., Bousquet, O., Lugosi, G., 2005. Theory of classification: a survey of recent advances. ESAIM Probability and Statistics To Appear.

Burges, C., 1998. A tutorial on support vector machines for pattern recognition. Data Mining Knowl. Discov. 2 (2), 1–32.

Calhoun, V., Adali, T., Pearlson, G., Pekar, J., 2001. Group ica of functional mri data: separability, stationarity, and inference. Proceedings ICA2001 (San Diego, CA).

Cox, D.D., Savoy, R.L., 2003. Functional Magnetic Resonance Imaging (fMRI) "brain reading": detecting and classifying distributed patterns of fMRI activity in human visual cortex. NeuroImage 19 (2), 261–270.

Dietterich, T.G., Bakiri, G., 1995. Solving multiclass learning problems via error correcting output codes. J. Artif. Intell. Res. 2, 263–286.

Efron, B., Tibshirani, R., 1998. An Introduction to the Bootstrap. Monogr. Stat. Appl. Probab., vol. 57 (Chapman and Hall).

Freund, Y., Schapire, R., 1996. A decision-theoritic generalization of on-line learning and an application to boosting. Proc. of the Ninth Annual Conference on Computational Learning Theory, pp. 325–332.

Gao, K., Posse, S., 2004. TurboFire: real-time fMRI with automated spatial normalization and Talairach Daemon database. NeuroImage 19 (2), 838.

Gembris, D., Taylor, J.G., Schor, S., Frings, W., Suter, D., Posse, S., 2000. Functional mr imaging in real-time using a sliding-window correlation technique. Magn. Reson. Med. 43, 259–268.

Haynes, J.D., Rees, G., 2005. Predicting the orientation of invisible stimuli from activity in human primary visual cortex. Nat. Neurosci. 8 (5), 686–691.

Hsu, C., Lin, C., 2001. A comparison on methods for multi-class support vector machines. Tech. rep., Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan.

Jeswani, S., Posse, S., 2004a. Fast determination of optimal classification spaces for fmri pattern classification (NeuroImage), WE203.

Jeswani, S., Posse, S., 2004b. Fast optimization of optimal classification spaces for fmri pattern recognition. Proc. of the 10th Meeting if the Organization for Human Brain Mapping, Budapest, Hungary.

Kamitani, Y., Tong, F., 2005. Decoding the visual and subjective contents of the human brain. Nat. Neurosci. 8 (5), 679–685.

Kiebel, S.J., Friston, K.J., 2004a. Statistical parametric mapping: I. Generic considerations. NeuroImage 2, 402–502.

Kiebel, S.J., Friston, K.J., 2004b. Statistical parametric mapping: II. A hierarchical temporal model. NeuroImage 2, 503–520.

Koltchinskii, V., Martínez-Ramón, M., Posse, S., 2005. Optimal aggregation of classifiers and boosting maps in functional magnetic resonance imaging. In: Saul, L.K., Weiss, Y., Bottou, L. (Eds.), Adv. Neural Inf. Process. Syst., vol. 17. MIT Press, Cambridge, MA, pp. 705–712.

LaConte, S., Strother, S., Cherkassky, V., Hu, X., 2003. Predicting motor tasks in fmri data with support vector machines. ISMRM Eleventh Scientific Meeting and Exhibition, Toronto, Ontario, Canada.

LaConte, S., Strother, S., Cherkassky, V., Anderson, j., Hu, X., 2005. Support vector machines for temporal classification of block design fmri data. NeuroImage 26, 317–329.

Magnotta, V.A., Harris, G., Andreasen, N.C., Yuh, W., Heckel, D., 2002. Structural mr image processing using the brains2 toolbox. Comput. Med. Imaging Graph. 26, 251–264.

Martínez-Ramón, M., Koltchinskii, V., Heileman, G., Posse, S., 2005. Pattern classification in functional mri using optimally aggregated ada-boosting. Organization of Human Brain Mapping, 11th Annual Meeting, Toronto, Canada, p. 909.

Martínez-Ramón, M., Koltchinskii, V., Heileman, G., Posse, S., 2005. Pattern classification in functional mri using optimally aggregated adaboost. Proc. International Society for Magnetic Resonance in Medicine, 13th Scientific Meeting, Miami, FL, USA.

Mason, L., Baxter, J., Bartlett, P., Frean, M., 1999. Advances in large margin classifiers. Ch. Functional Gradient Techniques for Combining Hypotheses. MIT Press, Cambridge, pp. 33–58.

Platt, J., Cristianini, N., Shawe-Taylor, J., 2000. Advances in neural information processing systems. Ch. Large Margin DAG's for Multi-class Classification, vol. 12. MIT Press, Cambridge, MA, pp. 547–553.

Posse, S., Binkofski, F., Schneider, F., Gembris, D., Wiese, S., Kiselev, V., Graf, T., Elghawaghi, B., Eickermann, T., 2001. A new approach to measure single event related brain activity using real-time fmri: feasibility of sensory, motor, and higher cognitive tasks. Hum. Brain Mapp. 12 (1), 25–41.

Posse, S., Fitzgerald, D., Gao, K., Habel, U., Rosenberg, D., Moore, G.J., Schneider, F., 2003. Real-time fmri of temporo-limbic regions detects amygdala activation during single trial self-induced sadness. NeuroImage 18, 760–768.

Rojo, J.L., Arenal, A., Artés, A., 2002. Discriminating between supraventricular and ventricular tachycardias from EGM onset analysis. IEEE Eng. Med. Biol. Mag. 21 (1), 16–26.

Schapire, R., 1999. A brief introduction to boosting. Proc. of the Sixteenth Intl. Conf. on Artificial Intelligence. Morgan Kauffman Pub. Inc., San Francisco, CA, USA, pp. 1401–1406.

Talairach, J., Tournoux, P., 1988. Co-Planar Stereotaxic Atlas of The Human Brain. Thieme, New York.

Vapnik, V., 1998. Statistical Learning Theory, Adaptive and Learning Systems for Signal Processing, Communications, and Control. John Wiley and Sons.

Wang, X., Hutchinson, R., Mitchell, T.M., 2004. Training fmri classifiers to discriminate cognitive states across multiple subjects. In: Thrun, S., Saul, L., Schölkopf, B. (Eds.), Adv. Neural Inf. Process. Syst., vol. 16. MIT Press, Cambridge, MA.

Weston, J., Watkins, C., 1999. Multi-class support vector machines. In: Verleysen, M. (Ed.), Proc. ESANN99. D-facto, Brussels, Belgium.